〔19〕中华人民共和国专利局

〔51〕Int.Cl.⁴
G11B 27/10

〔12〕发明专利申请公开说明书

〔11〕CN 88 1 00793 A

〔43〕公开日 1988 年 10 月 19 日

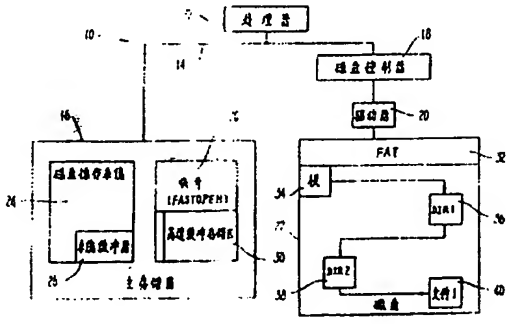
[21]申请号 88 1 00793
[22]申请日 88.2.27
[30]优先权
[32]87.3.30 [33]US [31]032,210
[71]申请人 国际商用机器公司
地址 美国纽约州
[72]发明人 肯尼斯·沃特·克里斯托福
巴里·阿兰·弗根包姆金·基姆
道格拉斯·卡莱顿·洛夫

[74]专利代理机构 中国国际贸易促进委员会专利代理部
代理人 王以平

[54]发明名称 快速开启由路径名识别的磁盘文件的方法

[57]摘要

一个数据处理系统具有以树型结构目录和文件的形式存贮在磁盘上的文件。该系统通过访问在主存贮器中某个驱动器高速缓冲存贮区而进行操作，以快速打开这些近期内已打开过或其部分路径信息有效的文件。高速缓冲存储区具有以树型结构链接的登记项，通过查寻这个树型结构，可以在打开文件过程中提供与否则必须由磁盘上得到信息相同信息。当高速缓冲存储区填满后，可用一个新项替换当前最少使用的项。



权 利 要 求 书

1．操作一个数据处理系统的方法，用于快速打开一个近期开过但目前已关闭的文件，该文件存贮在磁盘上并位于树型结构的目录和文件中，其特征在于下述步骤：

当上述文件首次打开时，在上述处理系统的主存储器中生成一个高速缓冲存储区，它包含许多用于查寻上述文件的链接的登记项位于通向该文件路径上的每个包含在上述的树型结构中的目录，上述文件的上述登记项包含从磁盘读出的信息，以及识别上述文件和确定文件位于上述磁盘中位置的信息；以及

为响应一个要求打开上述文件的连续请求，查寻在上述高速缓冲存储区中上述链接登记项并访问上述信息，由此避免为得到相同信息而必须从上述磁盘内读出上述信息的过程。

2．按照权利要求1的方法，其进一步的特征在于：上述数据处理系统有多个磁盘驱动器，并且上述方法包括：

生成与磁盘驱动器数目相同的多个高速缓冲存储区，并在每个高速缓冲存储区中设置一个用于识别每个驱动器的驱动器标题；

上述查寻步骤包括：首先至少访问一个驱动器标题，以确定与那个含有待打开的文件的驱动器相对应的高速缓冲存储区。

3．根据权利要求1的方法，其进一步特征在于：上述生成过程包括提供一个大小包含预定数目登记项的高速缓冲存储区；

最初填充上述缓冲存储区是将各登记项放置在其中未用的位置而实现的；之后，增设新项是将每个新登记项放到上述高速缓冲存储区中，对应于当前最少使用的文件的那个现存项的位置。

快速开启由路径名识别的磁盘文件的方法

本发明涉及有关数据处理的领域，尤其是涉及一种快速打开由路径名识别的磁盘文件的方法。

本发明旨在对众所周知并已商业化的这类系统进行改进，例如IBM个人计算机，在这类系统中，硬件是在磁盘操作系统(DOS)的管理下操作的，并且磁盘文件采用树型结构目录和路径名的存贮在硬磁盘中。信息按照柱面和扇区的预定格式存贮在磁盘上，每个扇区包含预定的字节数。为了访问所期望的扇区，磁头必须首先移至包含该扇区的柱面上，转动磁盘，使之经过磁头，直到磁头到达所需的扇区，才能读出该扇区的内容并将它放在缓冲区中。检查为访问磁盘上的数据所需的总时间可以看出，主要延时发生在磁头实际移动的过程中，在某一应用需要运行大量实际I／D设备的情况下，希望尽可能地减少磁头移动的次数。

文件按一簇或多簇扇区存贮在磁盘上，每个簇包含预定的扇区数。每个簇有一个唯一不同的起始地址。磁盘上文件的位置是借助于文件分配表(FAT)记录的，文件分配表本身存贮在该磁盘上。FAT的每个位置与不同的簇相关并且包含一个登记项，该登记项用来指示不存在其它 与某一文件相关或指向该文件的下一个簇号。短的文件包含在一个簇中。长文件则包含在若干链接在一起的族中。

文件通过使用树型结构目录来确定位置。每个磁盘包含一个根目录，若干子目录和大量文件。一个给定文件可以位于经过根目录和若干子目录的路经的尾部。每个目录包含一个用于附加目录和文件的登

记项。一个特定文件可以通过定义驱动器、路径和文件名来识别。例如：C：／ＤＩＲ１／ＤＩＲ２／ＦＩＬＥ１识别列在目录ＤＩＲ２中的文件名ＦＩＬＥ１，ＤＩＲ２是ＤＩＲ１的子目录，并列在其中，而ＤＩＲ１又是驱动器Ｃ的根目录的一个子目录，也列在其中。

当打开一个文件时，必须访问驱动器并搜寻这个路经上指定的所有目录，确定包含该文件名登记项的目录。再在这个目录里，搜寻所有的登记项和文件名直到找到所需要的文件为止。如果某个文件以前没有打开过，则没有登记项，因此在使用该文件前，必须设置登记项。如果该文件以前打开过，则含有该文件名的目录中的登记项又包含着一个登记项，该登记项是指向ＦＡＴ（对应着该文件起始位置的　）的索引。在这种打开过程中，为访问根目录，每个子目录和搜寻一长例文件名，必须有实际Ｔ／Ｏ设备参与。在某些应用中，在运行给定程序的过程中，相同的文件多次打开，并且每次打开文件沙及大量实际Ｔ／Ｏ设备的启用，因此能损失相当多的时间。本发明旨在改进现有开启文件的方法，即提供一种快速打开文件的方法并减少与开启文件过程相关的实际Ｉ／Ｏ设备的启用次数。

本发明的目的之一是提供一种在某文件最初已被至少打开一次之后，快速打开该文件的方法。

其另一目的是提供一种在高性能个人计算机系统中快速打开文件的方法，其具体作法是减少与开启文件过程有关的实际Ｉ／Ｏ设备的启用次数。

其目的之三是通过下述方法减少打开文件所需的时间，即，一旦该文件以前已经打开过，则去掉按通常方式为确定目录中的文件登记项而搜寻整个磁盘驱动器的目录以及不同文件名的过程，从而减少打

开文件的时间。

本发明的目的之四是提供快速再次打开那些近期已访问过的文件以及快速再次打开那些频繁开、闭的嵌套文件的方法。

简言之，做到这一点的方法是在高速缓冲存储区和主存贮器中建立并保持文件使用的历史。每当访问这一个目录或文件时，先检查其历史，如果打开文件所需的信息在高速缓冲存贮区中，则可直接使用，无需对可能涉及实际Ⅰ／Ｏ设备启用的任一目录或文件名进行树形搜索。通过保持高速缓冲存储区及主存贮器中的内容，这类信息就可以按主存贮器的高速度进行访问，而不涉及与磁盘的实际Ⅰ／Ｏ设备启用有关的较低速度。

通过下面参照附图所进行描述，本发明的其它目的及优点将是显而易见的，其中：

图１是一般性说明本发明与实现上述方法的环境及现有技术相互关系的简图。

图２是响应用户请求，按本发明所做的一般步骤流程图。

图３是（一个流程图，）用于选择各个具体过程的主过程流图。

图４是查寻过程流程图。

图５是插入过程的流程图。

图６是清除请求过程的流程图。

图７是删除过程的流程图。

图８是更新过程的流程图。

图９是说明ＤＯＳ如何调用本发明某些过程的流程图。

现在参考附图，首先看图１。数据处理系统１０（例如IBM AT个人计算机）包括：通过总线１４与主存储器１６和磁盘控制器１８

相连的处理器１２．磁盘控制器又依次与包括一个磁盘２２的磁盘驱动器２０相连，磁盘具有记录信息位的常规存储介质．在系统１０操作期间，ＤＯＳ过程２４装入或驻留在主存储器１６中，而它主要用于控制对磁盘２２上的信息的访问．ＤＯＳ与系统缓冲器２６相关联，即通过系统缓冲器使信息在磁盘２２和主存储器１６之间按通常方式进行传送。

还有一组叫作ＦＡＳＴＯＰＥＮ２８的过程也装入并驻留在主存储器１６中，该过程与ＦＡＳＴＯＰＥＮ高速缓冲存储区３０的操作有关，其细节将在下面描述。如果该系统中的驱动器多于一个，则将有对应数目的高速缓冲存储区３０，即每个存储区３０对应于一个驱动器。磁盘２２本身已格式化，以使其包含一个文件分配表３２（ＦＡＴ）和根目录３４。图１中还给出一个作为示例的文件结构，在该结构中，名为ＦＩＬＥ１的文件４０位于名为ＤＩＲ２的子目录３８中，ＤＩＲ２本身位于名为ＤＩＲ１的子目录３６中，而DIR1又是根目录３４的子目录。如果没有装入过程ＦＡＳＴＯＰＥＮ２８，则该系统按常规已知的方式操作。ＦＡＳＴＯＰＥＮ２８的装入通过下述过程实现，即对下面将介绍的各变量进行初始化，然后使其终止，但驻留在主存储器１６中，以便可根据需要进行调用。

如上所述，在该系统中每个磁盘驱动器对应有一个快速缓冲存储区３０。每个快速缓冲存储区包含一块称为驱动器高速缓冲存储区标题的信息。这个标题由若干不同字段组成，其意义如表１所示。

表1　　驱动器高速缓冲存储区标题

字段	意义
1	这个驱动器的ＬＲＵ链的标题
2	至ＬＲＵ最后一个登记项的偏移量
3	指向位于登记项链中第一个子项的指针
4	指向下一个驱动器高速缓冲存储区标题的指针
5	驱动器ＩＤ

在每个高速缓冲存储区３０中设有若干目录和文件登记项，每个登记项包含如表2所列的若干个信息字段。这些登记项组成一个链型数据结构，这个结构将在下面详细描述。高速缓冲存储区３０的尺寸将根据所希望的登记项数目或由用户所选的登记项数目来确定，上述所希望的登记项个数可通过系统省缺值选定。通常，由用户所选的数目应大于在驱动器中经登记项的最深嵌套。

表2　　登记项

字段	意义
1	指向下一个ＬＲＵ登记项的指针
2	指向下一个ＭＲＵ登记项的指针
3	指向子项的指针
4	指向兄弟项的指针
5	指向前一个结点的指针
6	目录／文件信息

较佳的作法是以下述方式将登记项维护在一个链中，这个方式是

当高速缓冲存储区３０充满登记项且需加入一个新项时，这个新登记项将取代高速缓冲存储区中当前最少使用项（ＬＲＵ）。因此，根据ＬＲＵ和ＭＲＵ（当前最频繁使用的项）的概念，用字段１和字段２将各登记链接起来登记项是针对每个已打开的文件以及每个含在通向该文件的路径中的子目录增设的。在同一子目录中的两个文件、两个目录或一个文件、一个目录认为是处于同一层，且称之为“兄弟”。其他子目录下的任一子目录或文件的登记项称为“儿子”。为了搜寻的目的，在高速缓冲存储区中的各登记项通过指针字段３－５以树型结构排列。字段６包含与存储在磁盘介质中的普通目录项有关的信息，除非该信息已插入到名字项的字段６中并因此驻留在主存储器里。因此，当需要打开一个在高速缓冲存储区３０中存有相应登记项文件时，这个信息能被快速访问。关于一个文件的信息包括：文件名和扩展名，属性，最后写入时间，最后写入日期，文件中第一个　号及文件尺寸。

参考图２，当一个应用或用户程序想打开某文件时，就要采用这个已知的用于打开文件的“ＤＯＳ调用中断２１Ｈｅｘ”。该调用进入ＤＯＳ２４，并且如前所指装入ＦＡＳＴＯＰＥＮ，以便可截取这个中断并对其发生作用。该过程的第一次判定是确定是否已装入ＦＡＳＴＯＰＥＮ，它由步骤５２实现。如果没有装入，则流程转向现有技术的步骤５４，即以通常方式访问磁盘来确定每个子目录和文件的位置，此后由步骤５６返回用户或应用。如果已装入FASTOPEN，则步骤５８按下面将详细描述的方式查寻请求。如果该过程出现错误，则从步骤６０分支到步骤５４中，并且该过程按照现有技术的过程进行处理。如果没有出现错误，则步骤６２确定该路径是部分被找到，或相反地已全部被找到。如果不是部分找到，即已找出整个路径以及

文件名，则得到和步骤５４所提供的相同信息，经由步骤５６返回至用户。如果仅找到部分路径，那么转入步骤６４，从已找出的信息点出发再继续访问磁盘以确定余下部分的路径和文件。在步骤６６中插入寻找到的各个部分，而步骤６８使该过程重复从步骤６４开始进行，直至到达该路径的尾部或找到文件名为止，同时将相应的信息返送给用户。

参考图3，这个过程的核心部分表示为“主（ｍａｉｎ）”，它用于确定应使用四个子过程中的哪一个过程。这四个子过程分别是“查寻”，“插入”，“删除”，和“更新”。这些不同的过程可以这样来调用，例如，通过把与之相对应的不同的数值插入处理器寄存器中的一个来实现。在任何情况下，步骤７０判定是否是查寻操作，同样地步骤７２、７４和７６分别判定是否是插入、删除或更新过程。如果这些过程均未调用，则返送回一个错误信息。下面根据各过程的特点，分别介绍这些过程。

参考图4，根据查寻过程，步骤７８首先确定与所申请的驱动器相关的高速缓冲存储区３０。然后步骤８０从预ＬＲＵ（ｐｒｅ－ＬＲＵ）堆栈中设置该ＬＲＵ链。这个堆栈是一个逻辑堆栈，用于在维护ＬＲＵ链时保持这个树型结构子目录。下一步，步骤８２形成一个回路的起始段，在这个回路中，现行结点是一个指针，它最初被设置指向该驱动器的首部，然后被设置指向现行结点的登记项。现行“兄弟”被设为零且开始查寻目录或文件名。遇到零值ＡＳＣⅡ码就确定为路径的结尾或文件名，由步骤８４判定这个过程。如果没有遇到零值ＡＳＣ码（这是在一次查寻刚刚开始时的一般情况），则执行步骤８６，使用指向现行登记项的子项的指针，从而找出现行结点下

的一个子项。步骤８８检查是否存在其它子项，即，该登记项在该序列中是否为最后一个子项？这个事实将由在子字段中的－１来指示。然后将期望的名字与该子项名字比较。如果由步骤９０进行比较的结果成功，则步骤９２生成一个ｐｒｅ－ＬＲＵ堆栈，将现行结点地址保存在该堆栈中并分支返回步骤８２。这个过程连续循环进行直到找出文件名，此时由步骤８４得到的“ｙｅｓ”或肯定结果则分支到步骤９４并将已找到的记录或信息返送回ＤＯＳ。如果步骤９０给出一个否定决断，则步骤９８查寻指向“兄弟”字段的指针并开始一个经步骤９６、９０和９８的循环过程，该循环直至无剩余“兄弟”时结束，则此时经步骤９４返回ＤＯＳ。当返回ＤＯＳ后，FASTOPEN将设置该指针指向从ＤＯＳ提供的路径／文件名字符串中找出的这个目录或文件名。ＤＯＳ又将依次查寻指针以便确定是否已找出整个路径，或者是找出部分路径或者什么也未找到。当存在较多“兄弟”时，步骤９６设置现行结点指针指向现行的“兄弟”。

参考图５，插入过程的目的是：基于现行结点和现行的“兄弟”信息，将目录或文件信息作为一个登记项插入到高速缓冲存储区中。步骤１０２由输入产生一个名字记录项，步骤１０４得到下一个可用项。在高速缓冲存储区未装满之前，一个新项可以简单地插入到任一位置，即登记项是自由的。一但高速缓冲存储区充满之后，则根据ＬＲＵ表得到的当前最少使用的项就成为要由新项替代的下一个可用项。步骤１０６把预ＬＲＵ堆栈清楚干净。如果现行“兄弟”指针等于零，则经步骤１１０将一个新项作为现行结点的一个子项插入，步骤１１２在预ＬＲＵ堆栈中记录该结点。如果步骤１０８中的现行“兄弟”不为零，则步骤１１４将一个新项作为现行结点的“兄弟”

插入

参考图6、图7，当某应用或用户请求删除一个特定文件时，则该过程不仅按通常方法删除磁盘中的该文件，而且删除高速缓冲存储区中与该文件有关的各登记项。因此，步骤116响应用户的请求，流程经DOS24由步骤118确定是否已装入FASTOPEN。如果没有装入，则步骤122访问磁盘并删除该文件，删除后经步骤124返回用户。如果已装入FASTOPEN，则发出删除请求120。为响应这个请求，步骤126查寻高速缓冲存储区以确定是否存在相应的登记项。如果没有找到该登记项，则简单地结束这个过程。如果找到该登记项，则通过步骤132将它从树型结构LRU链中移出并把此特定的项放在LRU链的顶部，作为可以使用的新项。首先删除一个文件项，然后删除那些不包含任何子项或兄弟项指针的任何“父 ”目录。

参考图8、图9，这里有某些位于FASTOPEN过程中并可由DOS直接调用的操作，用于完成某些附属于DOS所提供的功能的操作。DOS的请求附属于下述功能：即关闭文件、向文件里写一个零字节或产生一个节点，步骤142、144和146分别确定是否请求这些功能，然后分支到更新步骤134－140。由步骤136－140的肯定结果分别导致更新一个目录项，更新与某文件有关的簇号以及删除某个登记项。

在上述的讨论中，只谈到预LRU堆栈，现在介绍与LRU堆栈有关的内容，这两种堆栈均涉及使用位于登记项中的LRU指针和MRU指针。当该过程走过一条路径时，从根目录开始，经各子目录直至到达文件名，这对按照各结点的访问顺序生成预LRU堆栈。因

此，以由C：／DIR1／DIR2／FILE1访问或指定文件的例子中，各项的次序是根目录、DLR1、DIR2和FILE1。这种排列产生一个问题，即根目录应是当前最少使用的项假如重进行置换时就应删除该项，那么要访问某余的各登记项就无法进行了，一旦路径建立起来，通过使用Pre－LRU堆栈和重新将各项排序，从而使生成的LRU堆栈以文件名作为当前最少使用项。因此当要进行项的替换时，删掉的将是文件名，而不是象前面删掉任一双亲目录了。

显然，本领域的技术人员无需违反由权利要求书所规定的本发明的基本精神和范围，就可以进行其它变更和改进。

图.1

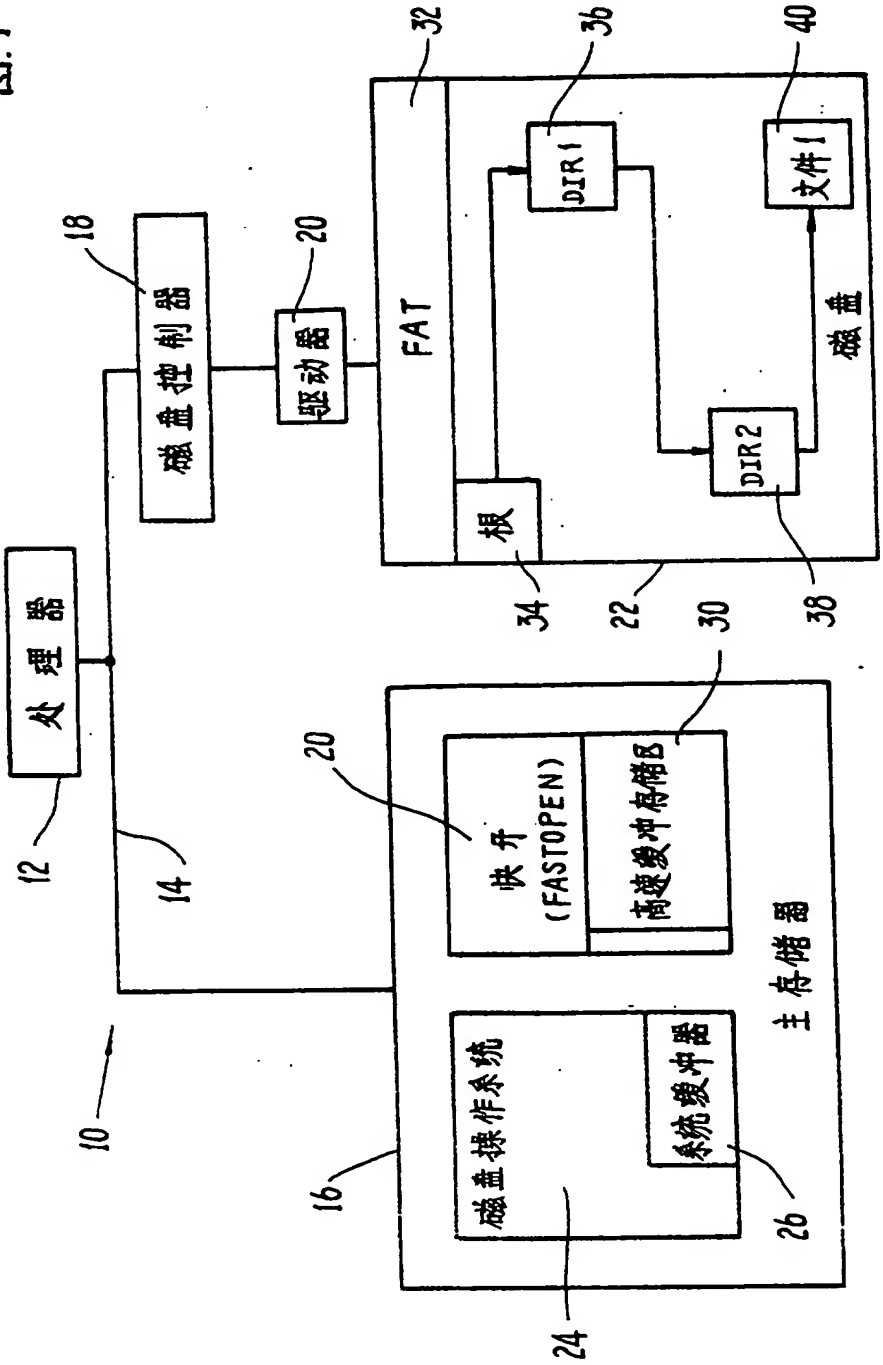

处理器
磁盘控制器
驱动器
FAT
根
DIR1
DIR2
文件1
磁盘
快开
(FASTOPEN)
高速缓冲存储区
磁盘操作系统
系统缓冲器
主存储器
10
12
14
16
18
20
22
24
26
30
32
34
36
38
40

图. 2

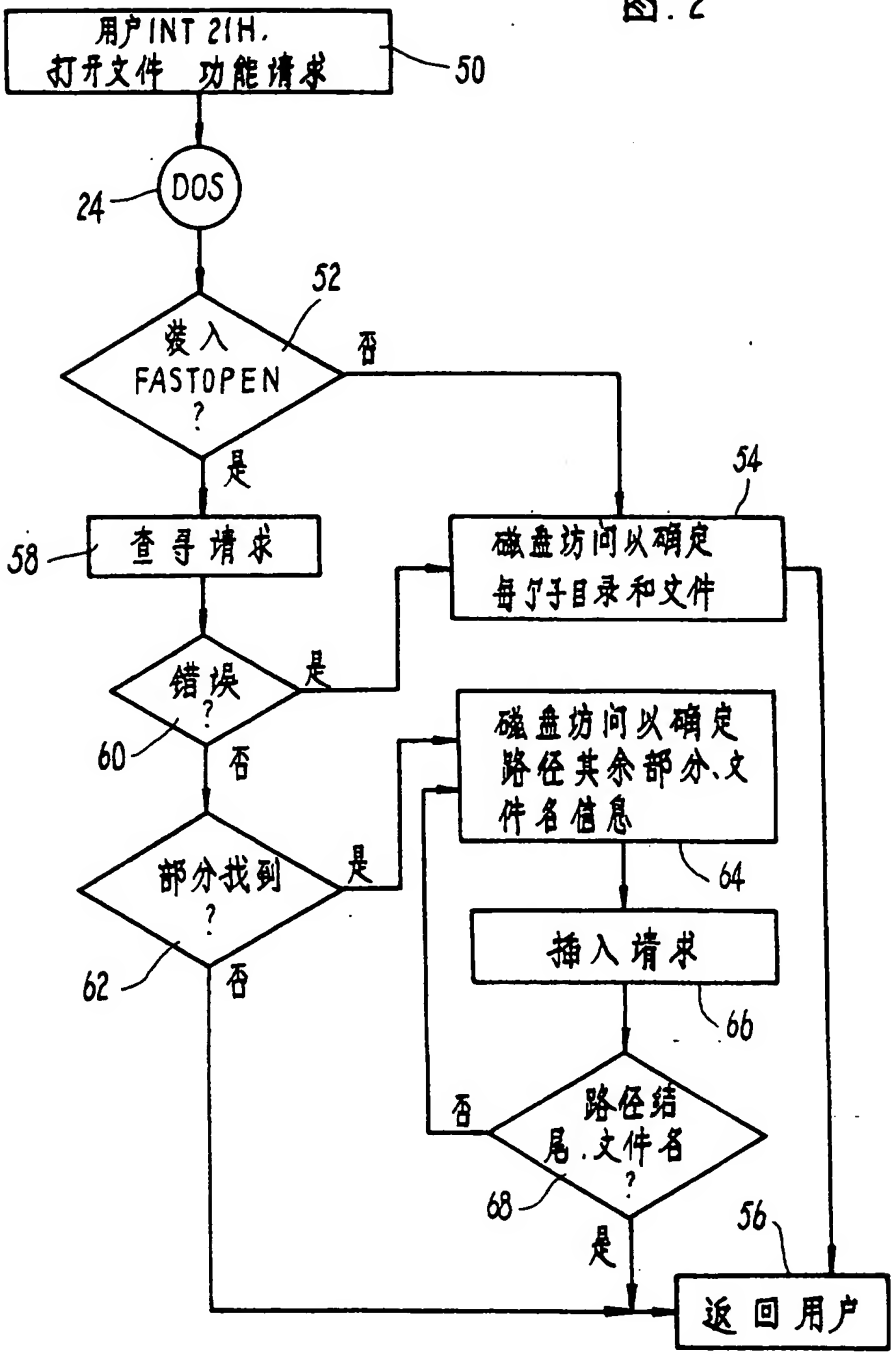
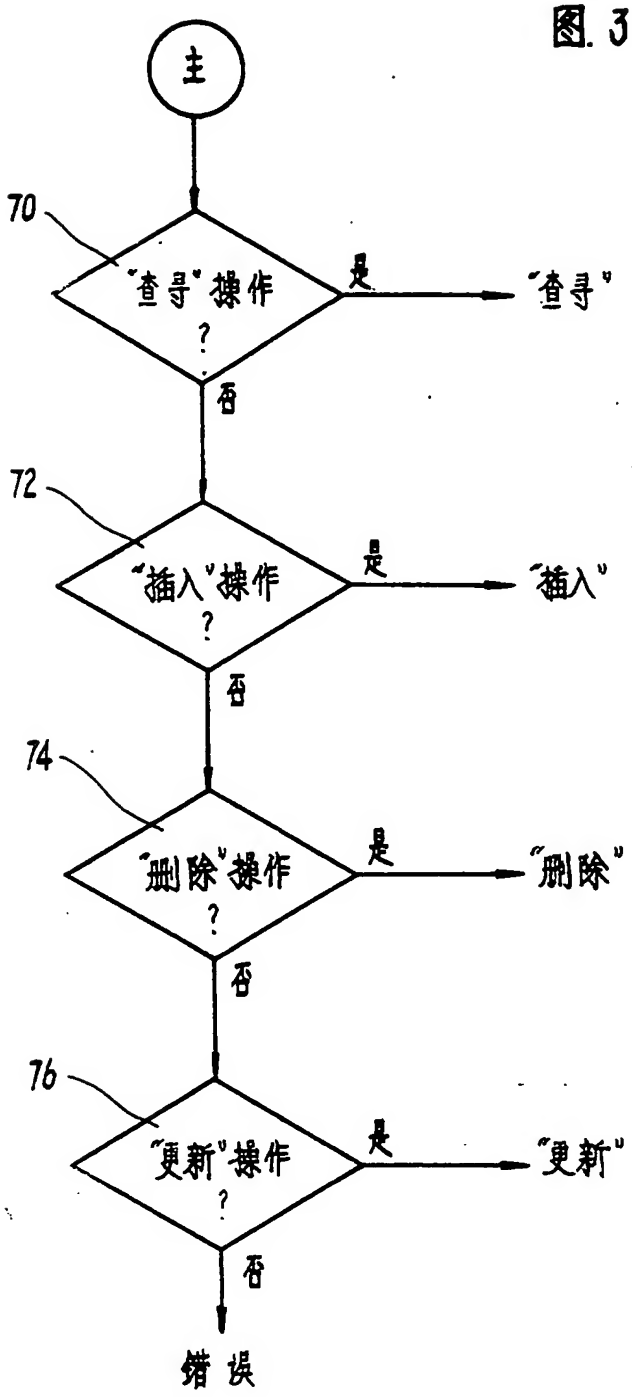

用户INT 21H.
打开文件 功能请求
50
DOS
24
52
装入
FASTOPEN
?
否
是
54
58
查寻请求
磁盘访问以确定
每个子目录和文件
错误
?
是
60
否
磁盘访问以确定
路径其余部分、文
件名信息
部分找到
?
是
64
62
否
插入请求
66
路径结
尾、文件名
?
否
68
56
是
返回用户

图. 3



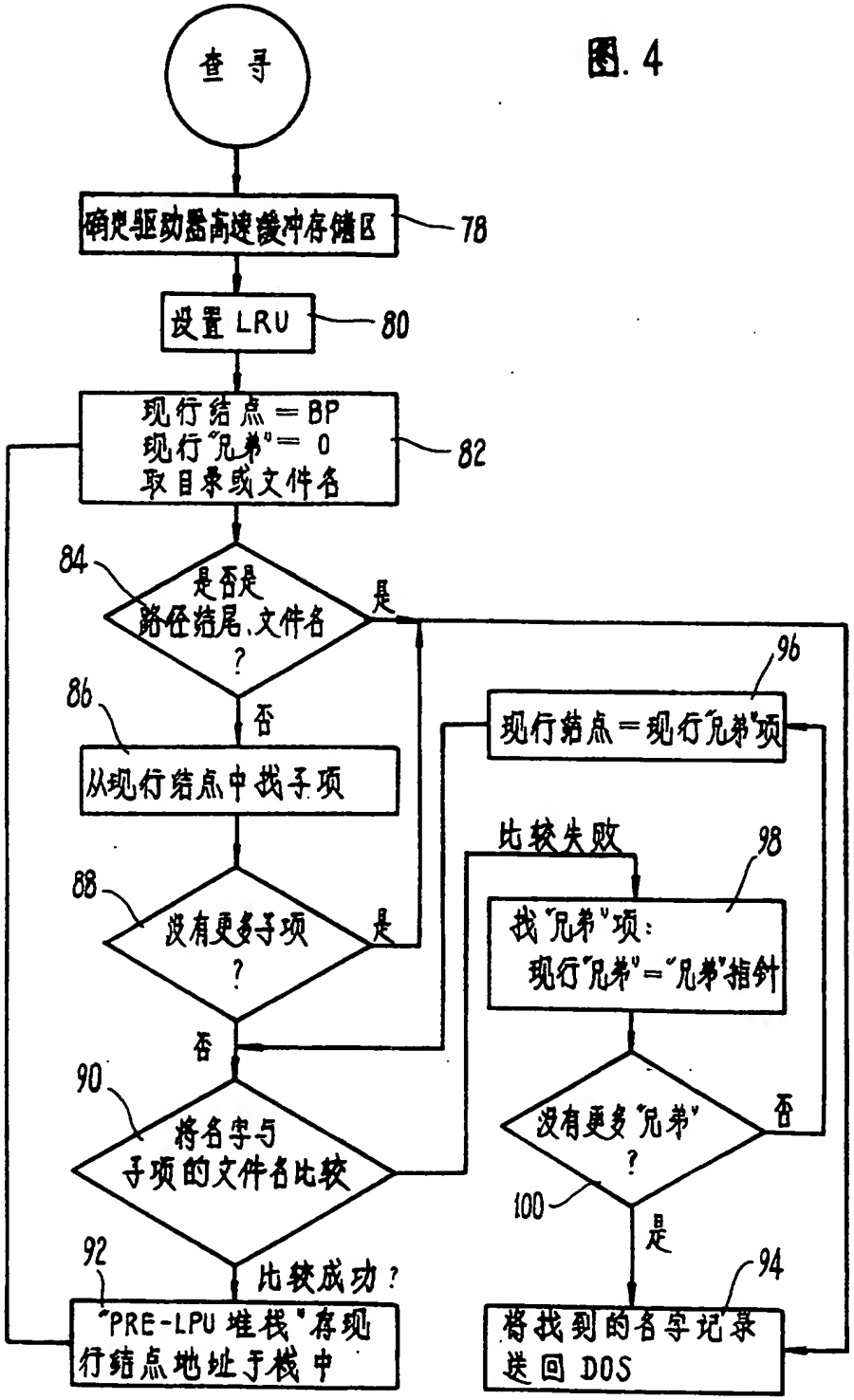
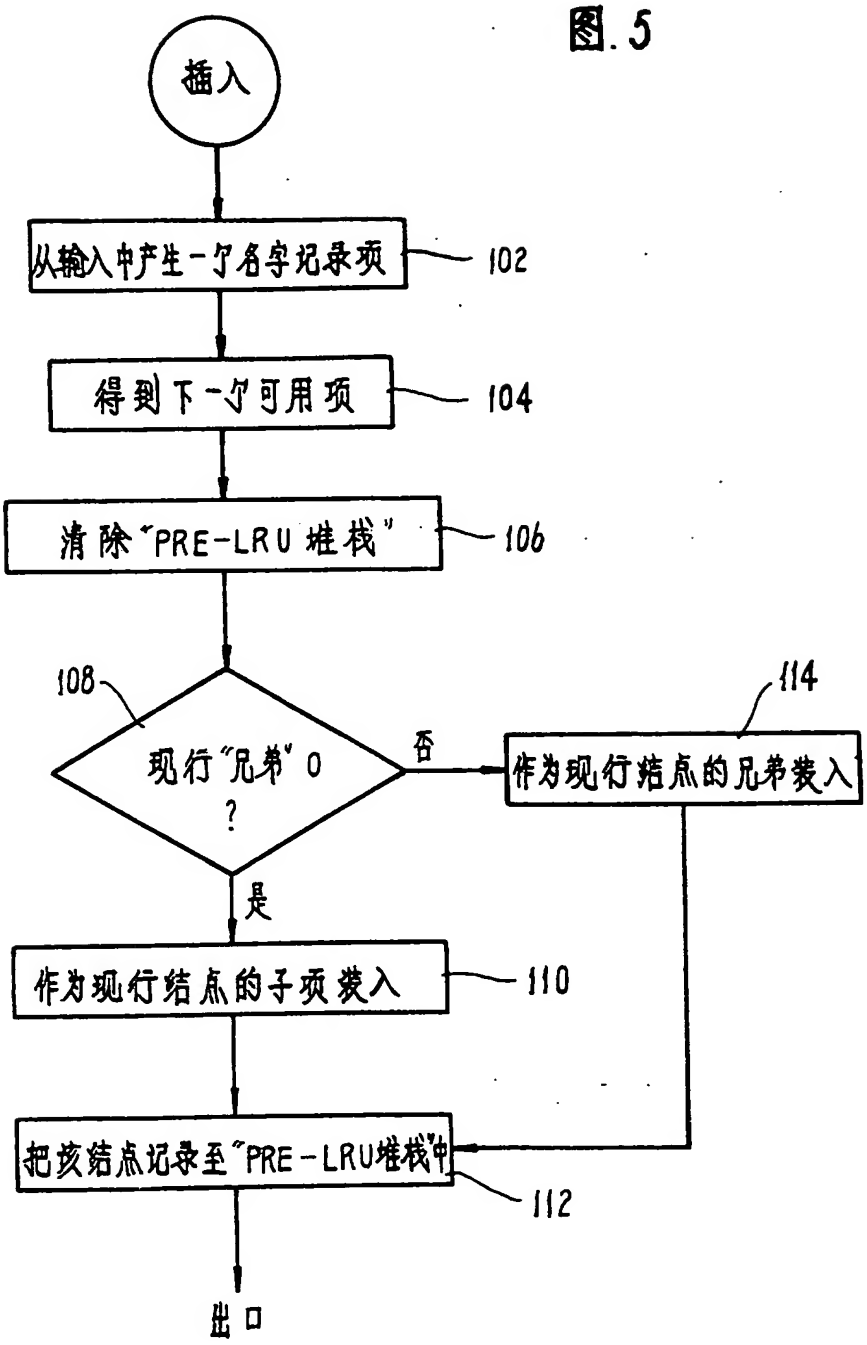

查寻
确定驱动器高速缓冲存储区
78
设置 LRU
80
现行结点＝BP
现行"兄弟"＝ 0
取目录或文件名
82
84
是否是
路径结尾.文件名
?
是
否
86
从现行结点中找子项
88
没有更多子项
?
是
否
90
将名字与
子项的文件名比较
比较失败
比较成功?
92
"PRE-LPU 堆栈"存现
行结点地址于栈中
96
现行结点＝现行"兄弟"项
98
找"兄弟"项:
现行"兄弟"＝"兄弟"指针
没有更多"兄弟"
?
否
100
是
94
将找到的名字记录
送回 DOS

图. 4

图.5


插入
从输入中产生一个名字记录项
102
得到下一个可用项
104
清除"PRE-LRU 堆栈"
106
108
现行"兄弟"0
?
否
114
作为现行结点的兄弟装入
是
作为现行结点的子项装入
110
把该结点记录至"PRE-LRU堆栈"中
112
出口

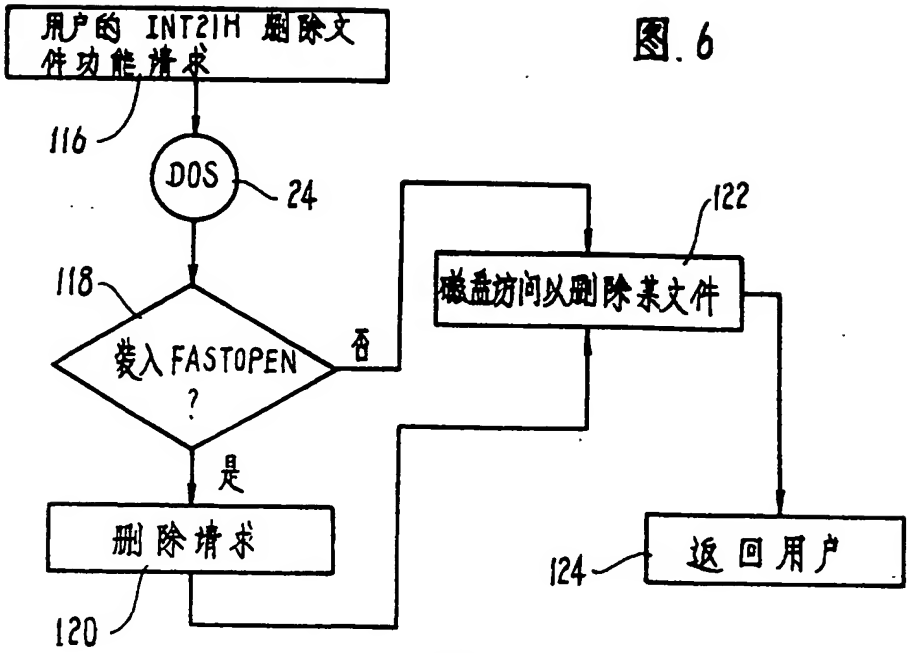

用户的 INT21H 删除文件功能请求
116
DOS
24
118
装入 FASTOPEN ?
否
是
122
磁盘访问以删除某文件
删除请求
120
返回用户
124

图.6

图.7

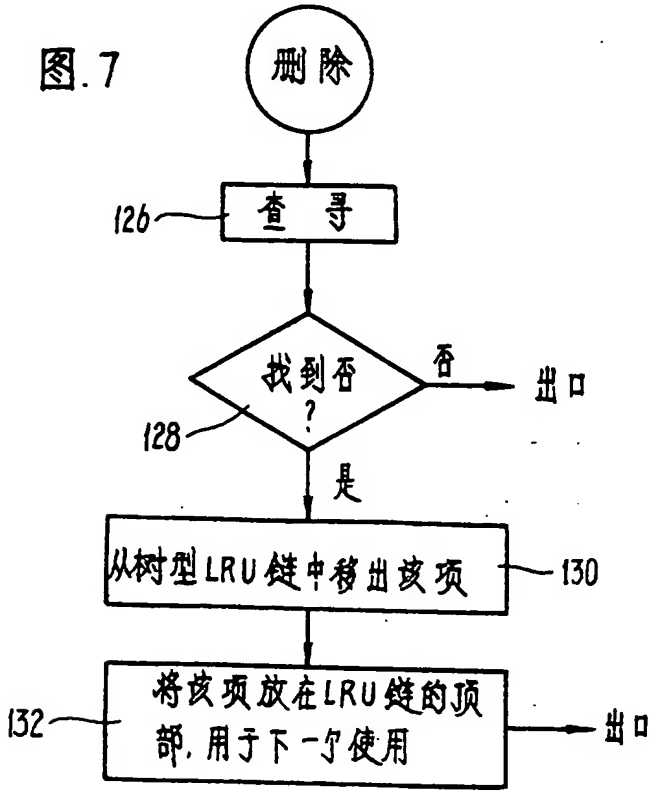

删除
126
查 寻
找到否 ?
128
否
出口
是
从树型LRU链中移出该项
130
将该项放在LRU链的顶部，用于下一个使用
132
出口

图. 8

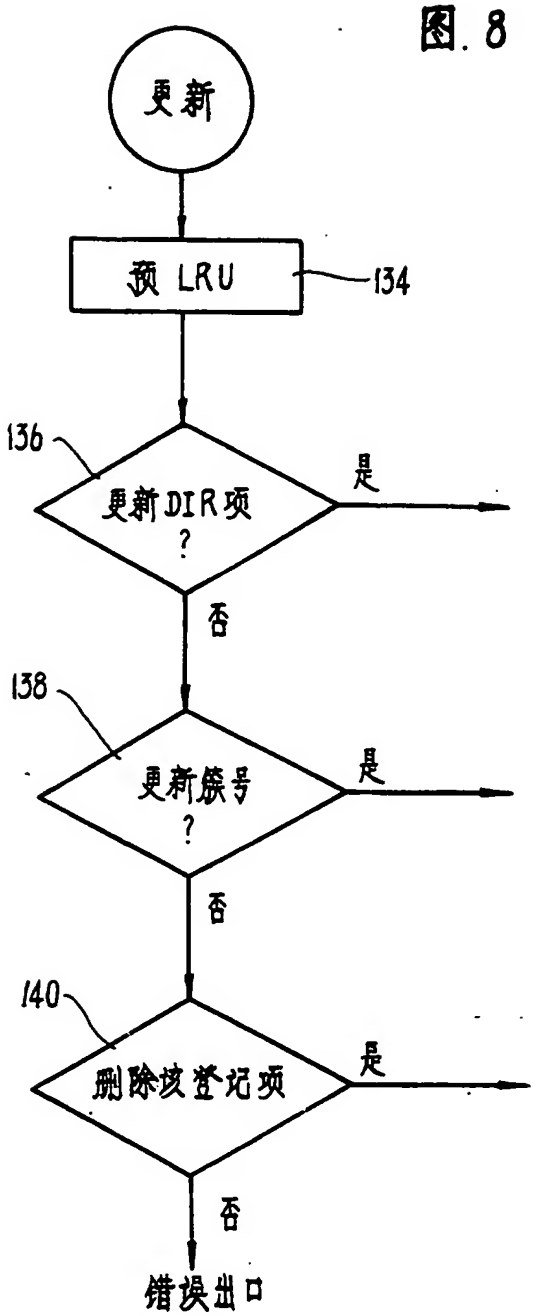
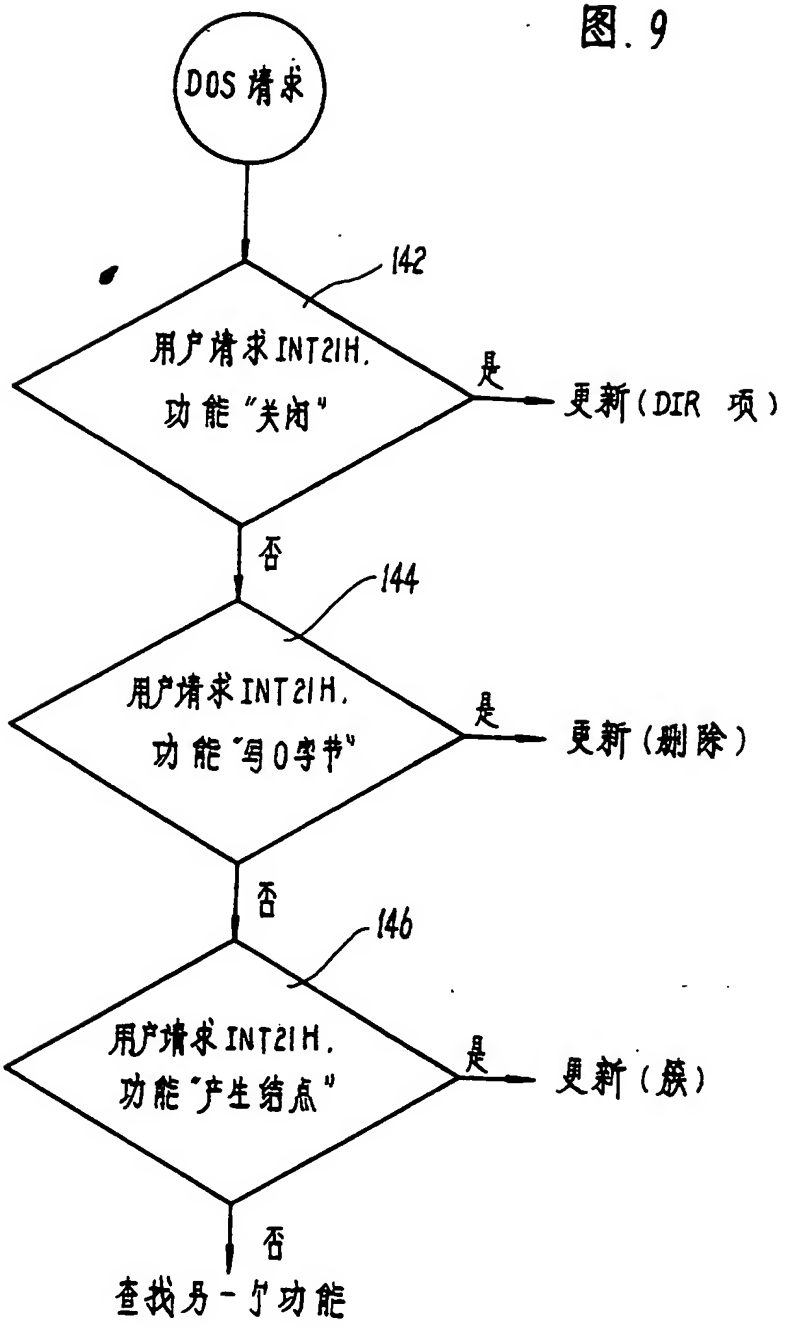


图. 9


DOS 请求
142
用户请求INT21H.
功能"关闭"
是
更新(DIR 项)
否
144
用户请求INT21H.
功能"写0字节"
是
更新(删除)
否
146
用户请求INT21H.
功能"产生结点"
是
更新(簇)
否
查找另一个功能